	<b>Kryptographie</b>	it.schule stuttgart Breitwiesenstrasse 20-22 70565 Stuttgart
	Übung – Signaturen, Hash-Werte und Zertifikate	

Nach Lösen von Aufgabe 2 „man in the middle“ des Übungsblattes „Hybride Verfahren“ könnte man zu dem Schluss gelangen, dass der Aufwand, den die Erfinder der Public-Key-Kryptographie getrieben haben leider vergeblich war.

Man könnte meinen, dass es reichlich sinnlos sei, zunächst über einen vertrauenswürdigen Kanal öffentliche Schlüssel zu verteilen - stattdessen könnten dann ja gleich symmetrische Schlüssel verteilt werden, die sich dann im Anschluss ohnehin besser für umfangreiche Datentransfers eignen.

Doch bei der dieser Argumentationskette handelt es sich um einen Trugschluss. Um zu verstehen weshalb, müssen wir Public-Key-Kryptographie nur einmal „verkehrt herum“ denken.

### **Aufgabe 1:**

Bisher hatten wir den öffentlichen Schlüssel (des Empfängers) benutzt, um Nachrichten zu verschlüsseln. Tatsächlich kann man aber auch seinen geheimen Schlüssel verwenden, um Nachrichten zu verschlüsseln, die dann jeder mit dem dazugehörigen öffentlichen Schlüssel entschlüsseln kann.

Tatsächlich wird dies in leicht abgewandelter Form in der Praxis sehr häufig getan. Überlegen Sie sich, was man damit bezwecken könnte.

### **Hash-Werte**

Wir hatten auf dem Arbeitsblatt über asymmetrische Kryptographie gelernt, dass sich das RSA-Verfahren nicht dazu eignet, große Dateien zu verschlüsseln. Wir benötigen daher ein weiteres kryptographisches Konstrukt, die sogenannten kryptographischen Hash-Werte.

Man kann zu jedem beliebigen Dokument einen eindeutigen kryptographischen Hash-Wert erzeugen. Verwendet man ein gutes kryptographisches Verfahren wie z.B. sha-256 oder md5, so kann niemand ein anderes Dokument generieren das denselben Hash-Wert aufweist (und erst recht keines, welches dem ursprünglichen Dokument ähnlich ist).

Hash-Algorithmen können auch für große Dokumente sehr schnell HASH-Werte erzeugen.

In Aufgabe 1 hatten wir ein Gedankenexperiment unternommen und haben erkannt, dass man die Verschlüsselung mit einem geheimen Schlüssel als eine Art „Unterschrift“ interpretieren könnte.

Das gleiche Ziel erreicht man, indem man nicht das Dokument selbst verschlüsselt, sondern den zugehörigen Hash-Wert des Dokuments. Dies hat außerdem die Vorteile:


- dass dies auch für große Dokumente funktioniert,
- dass das unterschriebene Dokument im Klartext vorliegen kann/soll
- und Dokument und Unterschrift voneinander getrennt (aufbewahrt) werden können.

### **Aufgabe 2:**

Erzeugen Sie mit dem folgenden Befehlen Hash-Werte für die Datei: einFilm.avi

```
openssl dgst einFilm.avi
```

```
openssl dgst -sha256 einFilm.avi
```

	<b>Kryptographie</b>	it.schule stuttgart Breitwiesenstrasse 20-22 70565 Stuttgart
	Übung – Signaturen, Hash-Werte und Zertifikate	

Erzeugen Sie mit den folgenden Befehlen ein Unterschrift für die Datei: einFilm.avi

```
openssl dgst -out einFilm.avi.sign -sign private.pem einFilm.avi
```

Überprüfen Sie diese Unterschrift mit dem Befehl:

```
openssl dgst -verify public.pem -signature einFilm.avi.sign einFilm.avi
```

## **Zertifikate und CAs:**

### **Aufgabe 3:**

Überlegen Sie sich, wie mit Hilfe digitaler Unterschriften das Problem der Schlüsselverteilung gelöst werden kann, bzw. auf ein erträgliches Maß reduzierbar ist.

### **Aufgabe 4:**

Welche große Gefahr kann bei der heute üblichen Verfahrensweise zur Verteilung von Schlüsselpaaren auftreten bzw. wo werden Hacker, Geheimdienste und weitere Interessierte wohl ansetzen?

Gibt es bereits Beispiele für solches Verhalten?